BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

FIG. 1

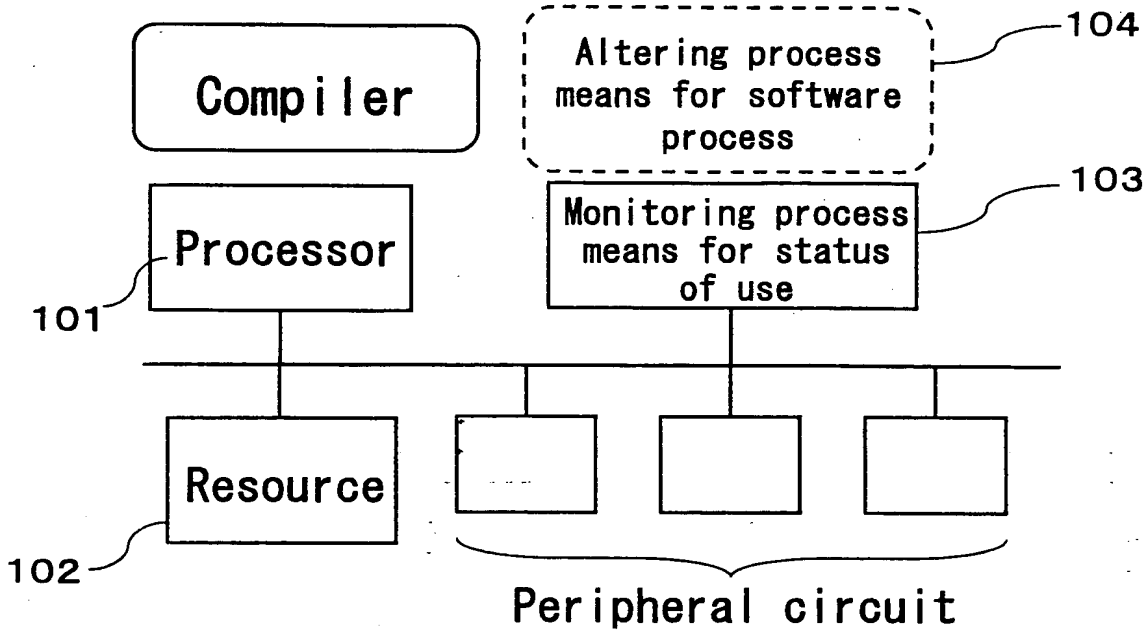


FIG. 2

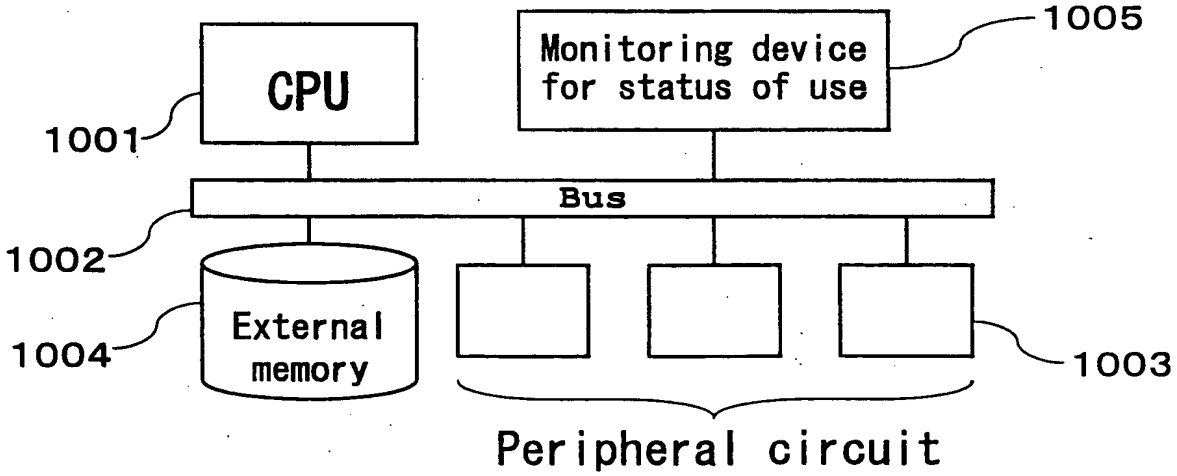


FIG. 3

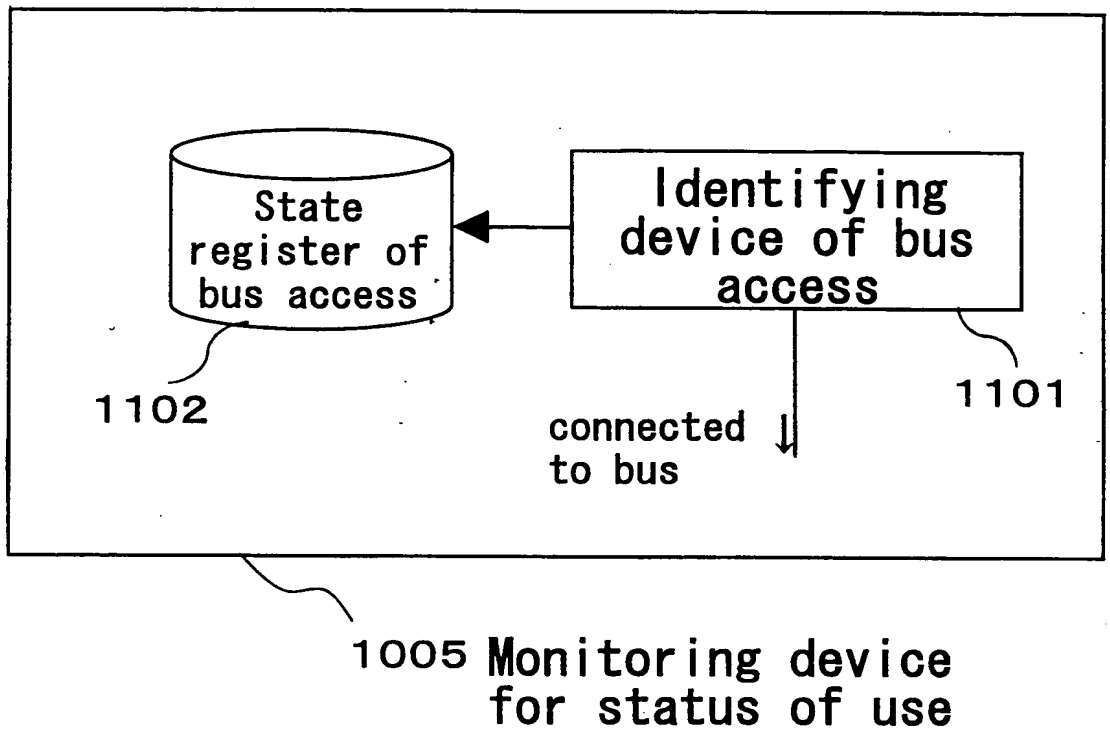

State register of bus access
Identifying device of bus access
1102
1101
connected to bus
1005 Monitoring device for status of use

FIG. 4A

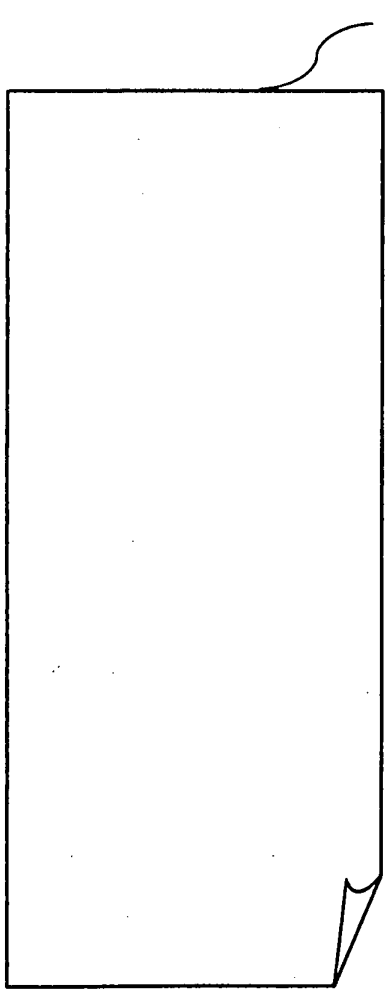


FIG. 4B

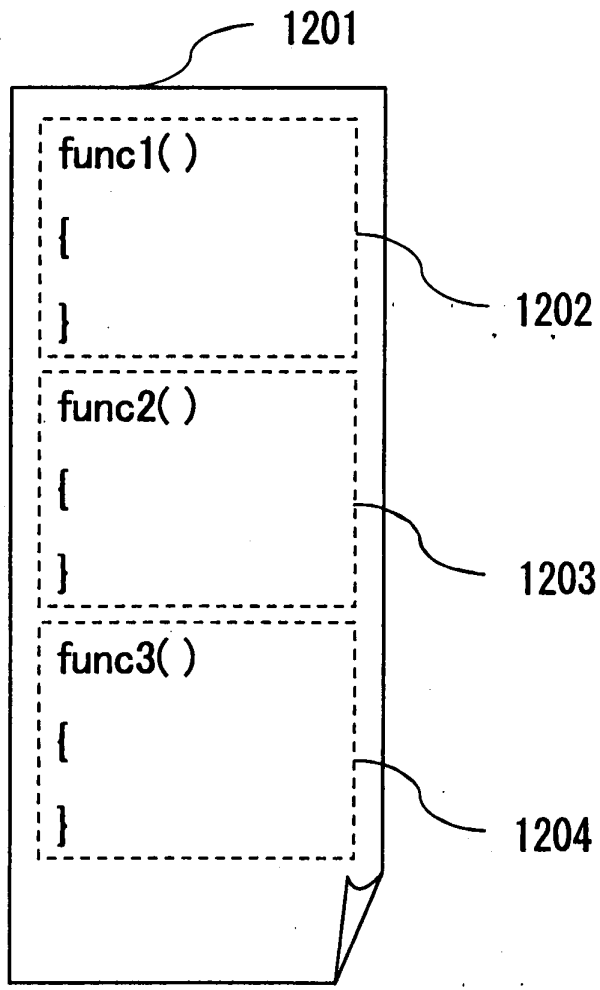


FIG. 4C

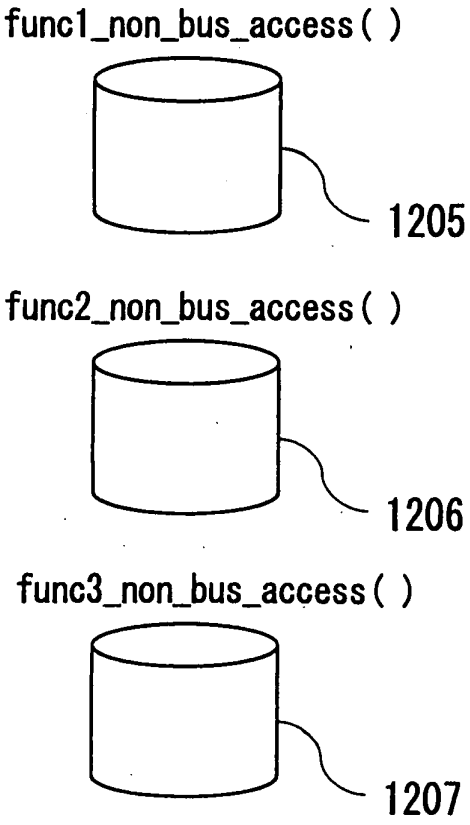


FIG. 5

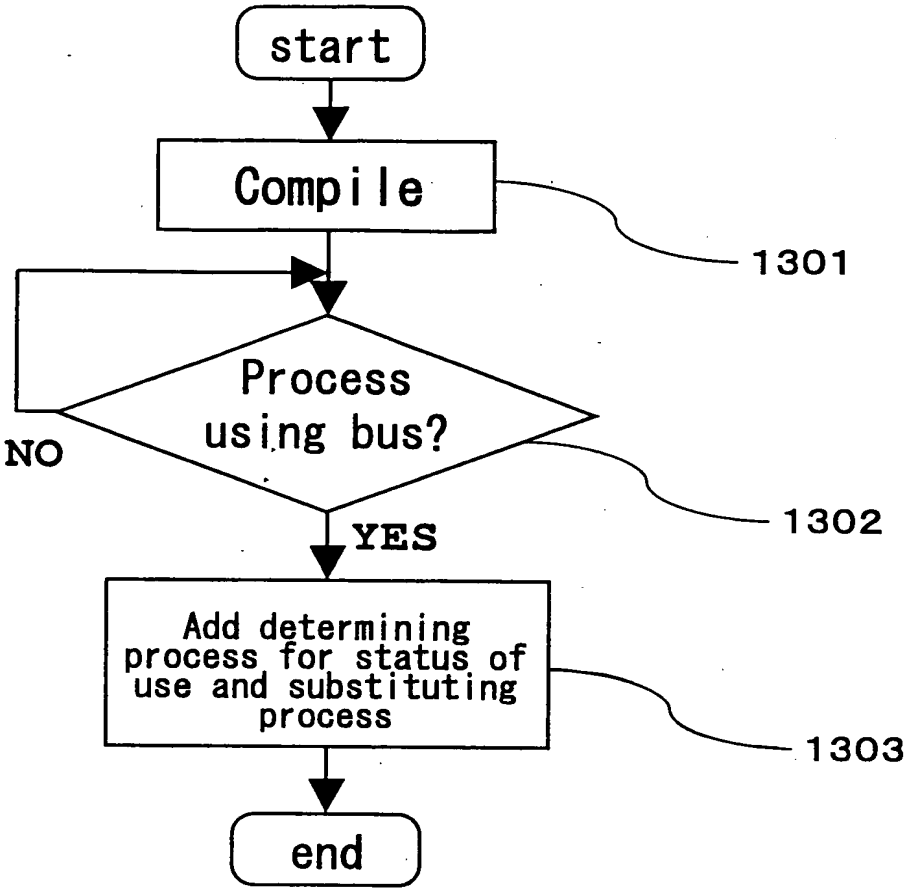

start
Compile
1301
Process using bus?
NO
YES
1302
Add determining process for status of use and substituting process
1303
end

FIG. 6

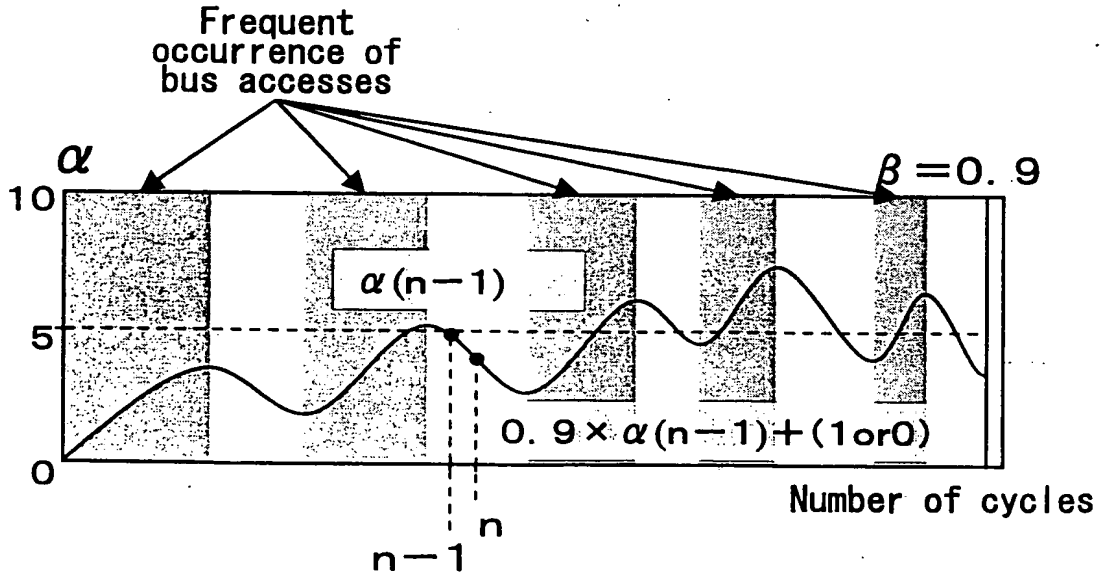

Frequent occurrence of bus accesses
α
β=0.9
10
5
0
α(n−1)
0.9×α(n−1)+(1or0)
Number of cycles
n−1
n

FIG. 7A

```
main( )
{
        :
        :
    Func1( )
        :
    Func2( )
        :
    func3( )
        :
        :
}
  :
  :
```

FIG. 7B

```
Func1( )
{
    if(bus access is occurring)
        func1_non_bus_access( );
    else
        func1( );
}
Func2( )
{
    if (bus access is occurring)
        func2_non_bus_access( );
    else
        func2( );
}
  :
```

FIG. 8

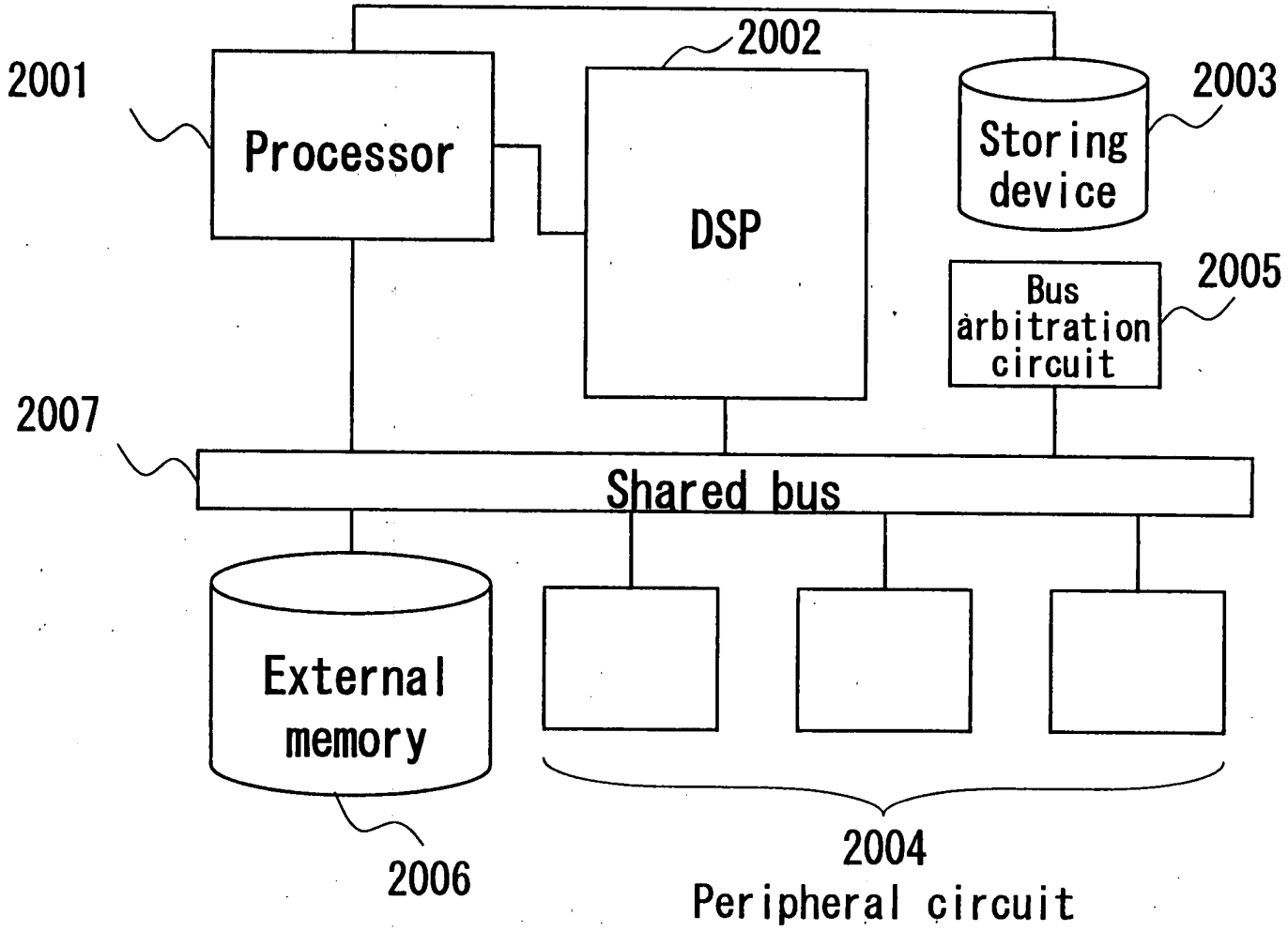

2001
Processor
2002
DSP
2003
Storing device
2005
Bus arbitration circuit
2007
Shared bus
External memory
2006
2004
Peripheral circuit

FIG. 9

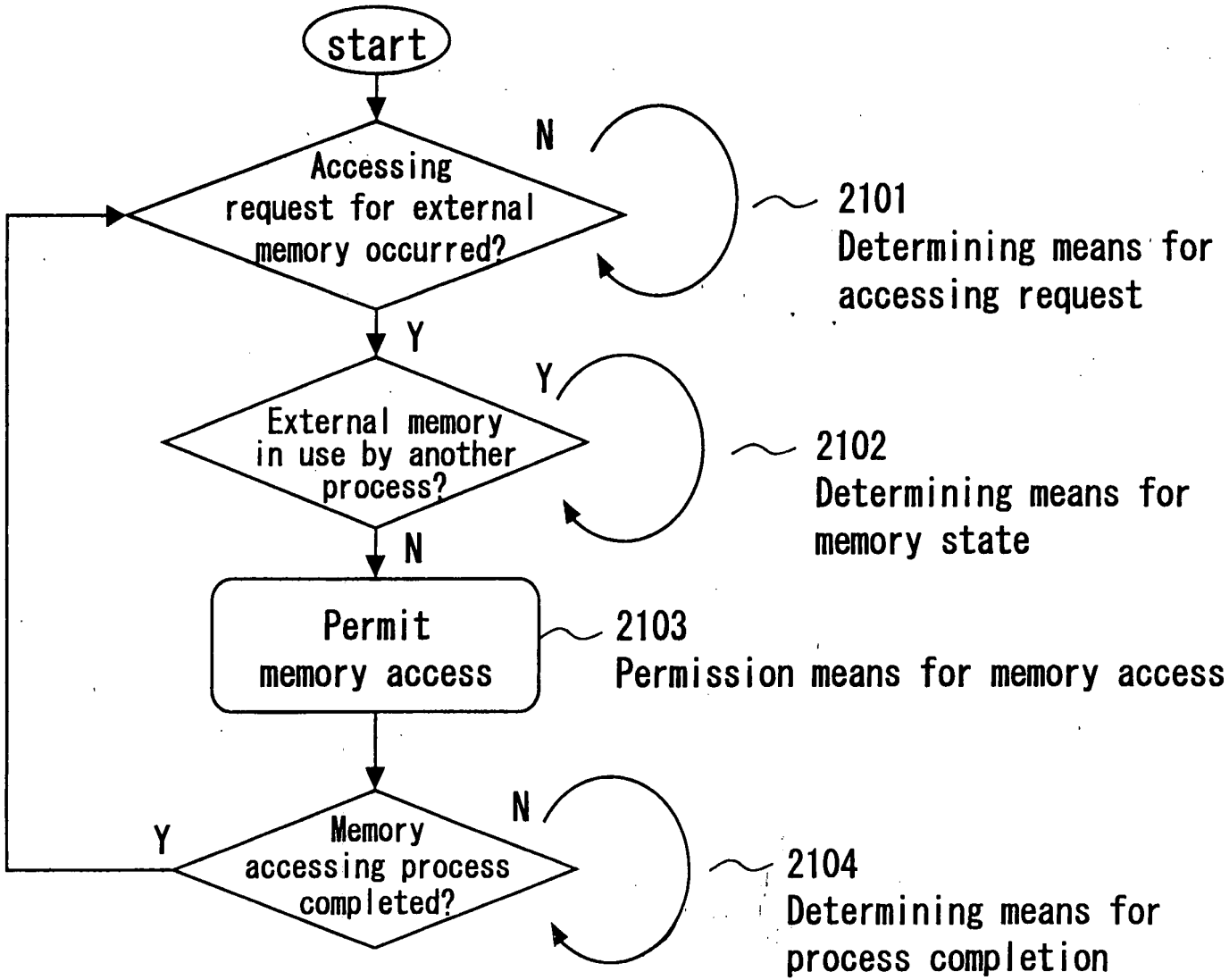

start
Accessing request for external memory occurred?
N
Y
2101
Determining means for accessing request
External memory in use by another process?
Y
N
2102
Determining means for memory state
Permit memory access
2103
Permission means for memory access
Memory accessing process completed?
N
Y
2104
Determining means for process completion

FIG. 10

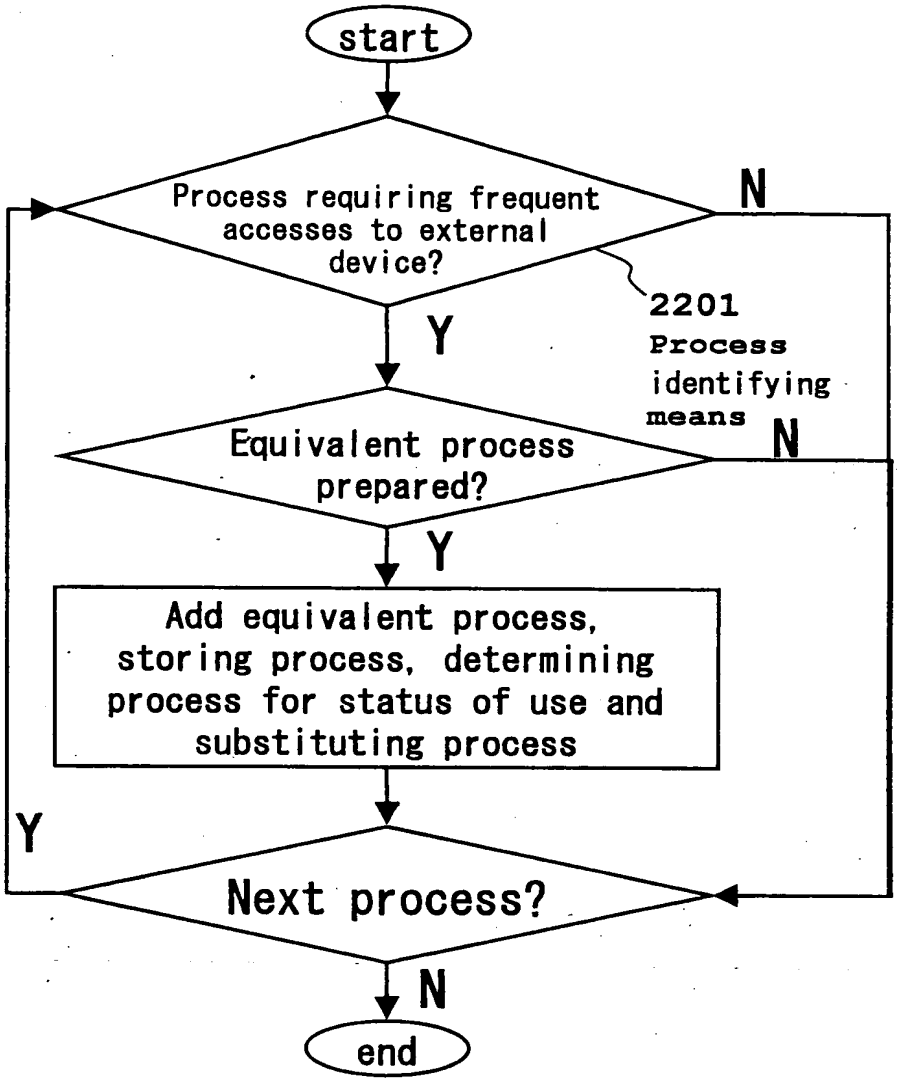

start
Process requiring frequent accesses to external device?
N
Y
2201
Process identifying means
Equivalent process prepared?
N
Y
Add equivalent process, storing process, determining process for status of use and substituting process
Y
Next process?
N
end

FIG. 11A

```
1: proc_task1(){
2:       mem_acs_many();
3: };
```

Prior to compiler execution

FIG. 11B

```
 1: proc_task1(){
 2:     if( task_time < DEFINED_TIME ) {
 3:         start_time = timer_count;
 4:         mem_acs_many();
 5:         end_time   = timer_count;
 6:         task_time = end_time - start_time;
 7:     }
 8:     else{
 9:         mem_acs_few();
10:     };
11: };
```

After compiler execution

FIG. 12

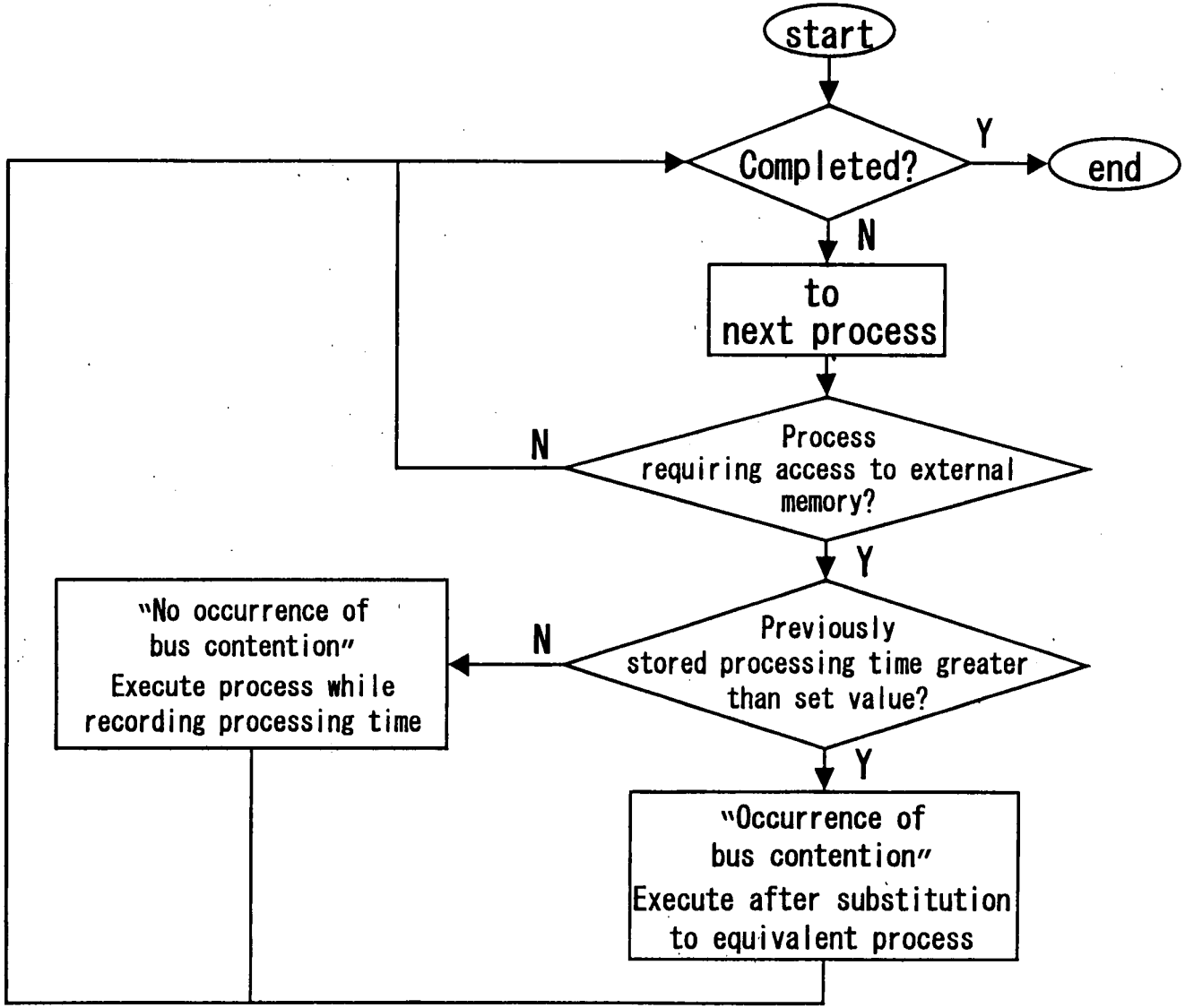

start
Completed?
Y
end
N
to
next process
N
Process
requiring access to external
memory?
Y
"No occurrence of
bus contention"
Execute process while
recording processing time
N
Previously
stored processing time greater
than set value?
Y
"Occurrence of
bus contention"
Execute after substitution
to equivalent process

FIG. 13

Processing time / Set value	Probability of substitution to equivalent process [%]	Probability of non-substitution to equivalent process [%]
-2.0	60	40
2.0 - 1.8	70	30
1.8 - 1.6	80	20
1.6 - 1.4	90	10
1.4 - 1.2	95	5
1.2 - 1.0	97	3
less than 1	0	100

FIG. 14

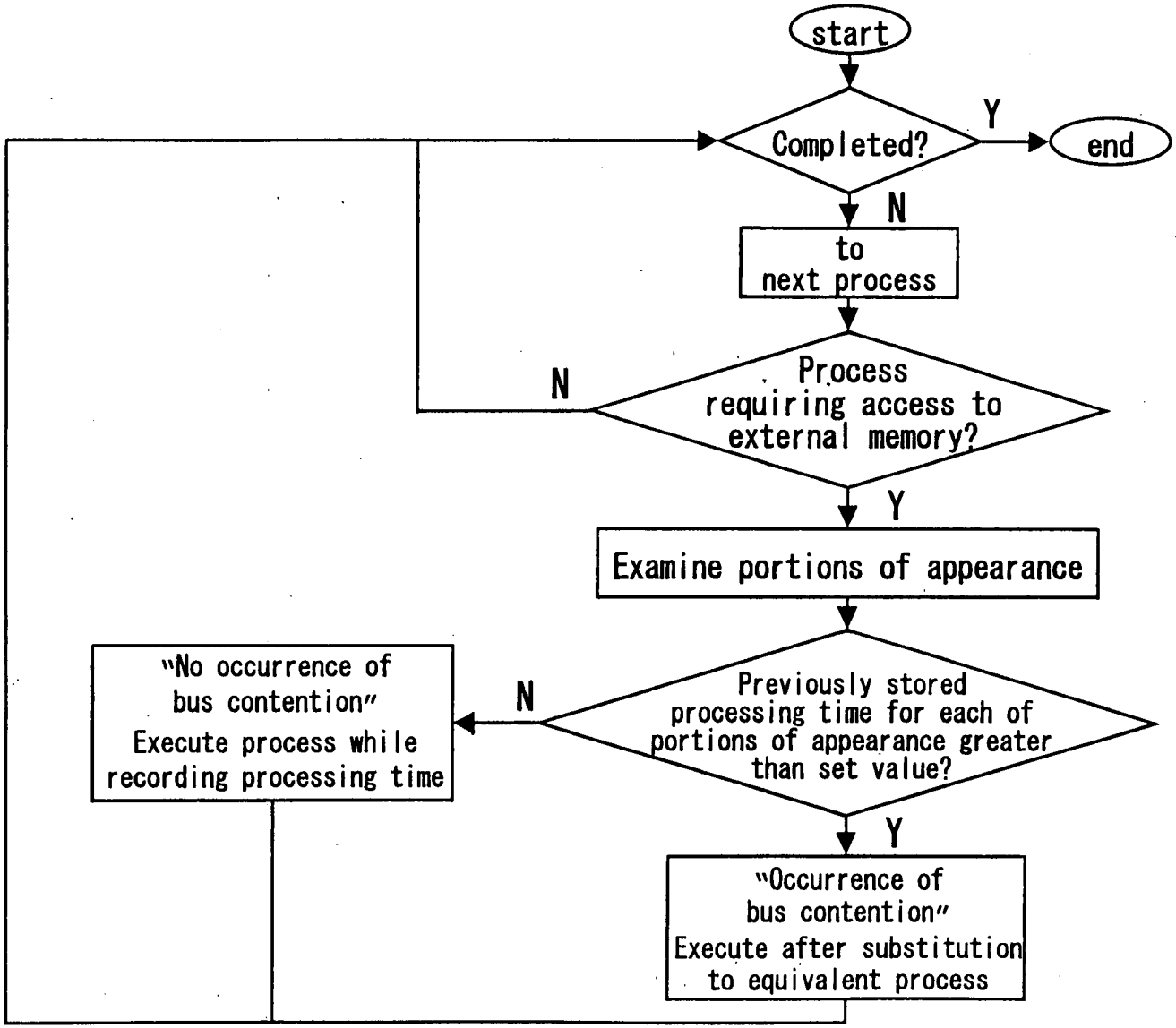

start
Completed?
Y
end
N
to
next process
Process
requiring access to
external memory?
N
Y
Examine portions of appearance
"No occurrence of
bus contention"
Execute process while
recording processing time
N
Previously stored
processing time for each of
portions of appearance greater
than set value?
Y
"Occurrence of
bus contention"
Execute after substitution
to equivalent process

FIG. 15

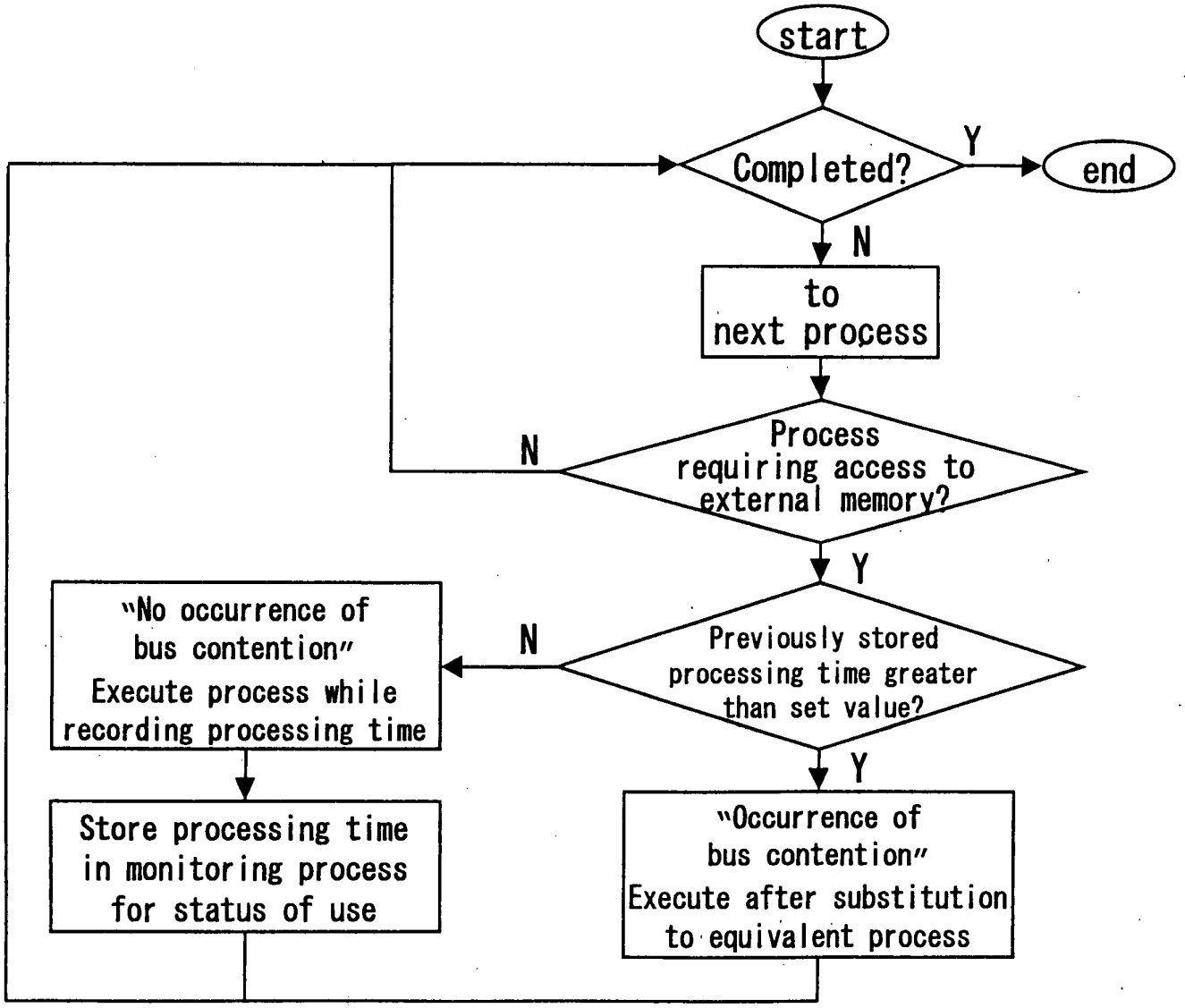

start
Completed?
Y
end
N
to
next process
Process
requiring access to
external memory?
N
Y
"No occurrence of
bus contention"
Execute process while
recording processing time
N
Previously stored
processing time greater
than set value?
Y
Store processing time
in monitoring process
for status of use
"Occurrence of
bus contention"
Execute after substitution
to equivalent process

FIG. 16

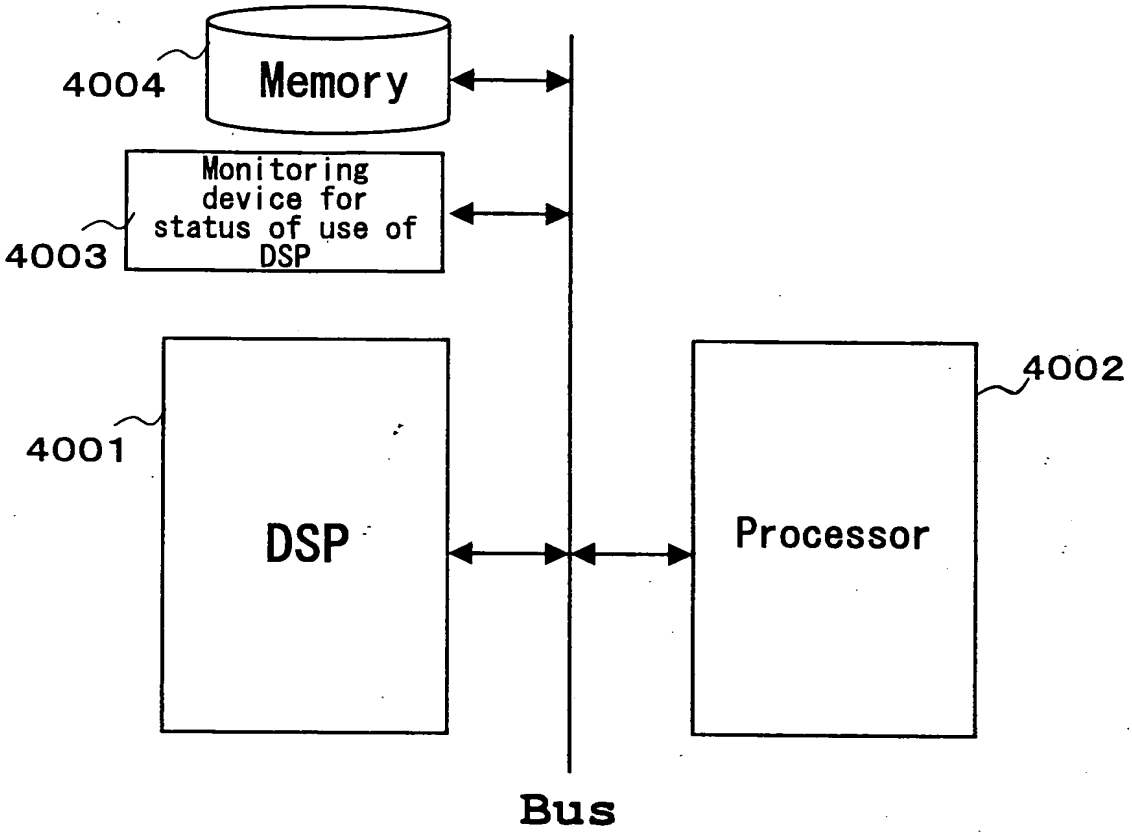

4004
Memory
Monitoring device for status of use of DSP
4003
4001
DSP
Processor
4002
Bus

FIG. 17

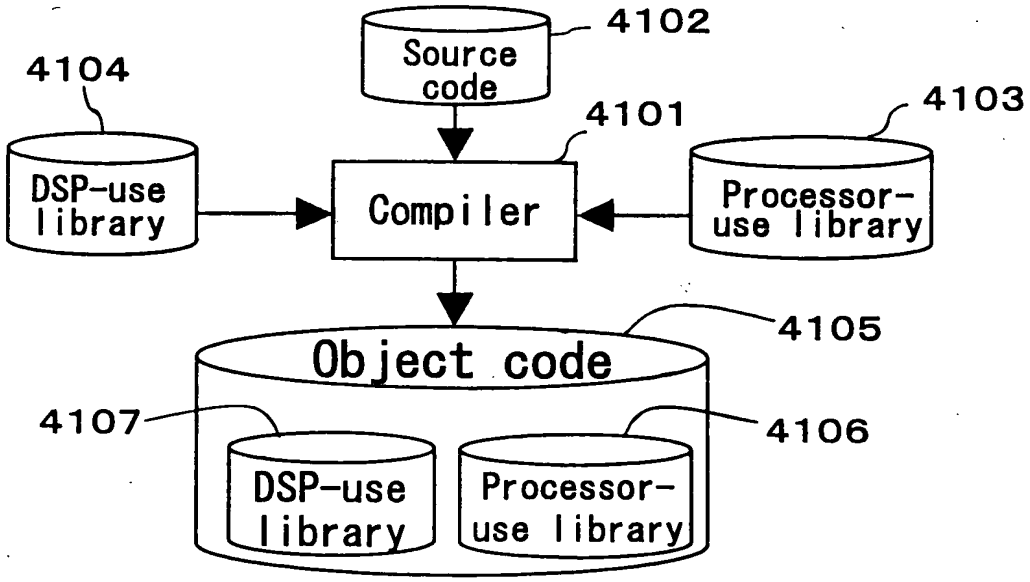

4102
Source code
4104
4101
4103
DSP-use library
Compiler
Processor-use library
4105
Object code
4107
4106
DSP-use library
Processor-use library

FIG. 18A

```
1:  main( )
2:  {
3:          :
4:      func1( )
5:          :
6:      func2( )
7:          :
8:      func3( )
9:          :
10: }
```

FIG. 18B

```
1:  main( )
2:  {
3:          :
4:      func1( )
5:          :
6:      func2( )
7:      {
8:        if(DSP is in waiting state)
9:              func2_dsp( );
10:         else
11:             func2_cpu( );
12:     }
13:         :
14:     func3( )
15:         :
16: }
```

FIG. 19

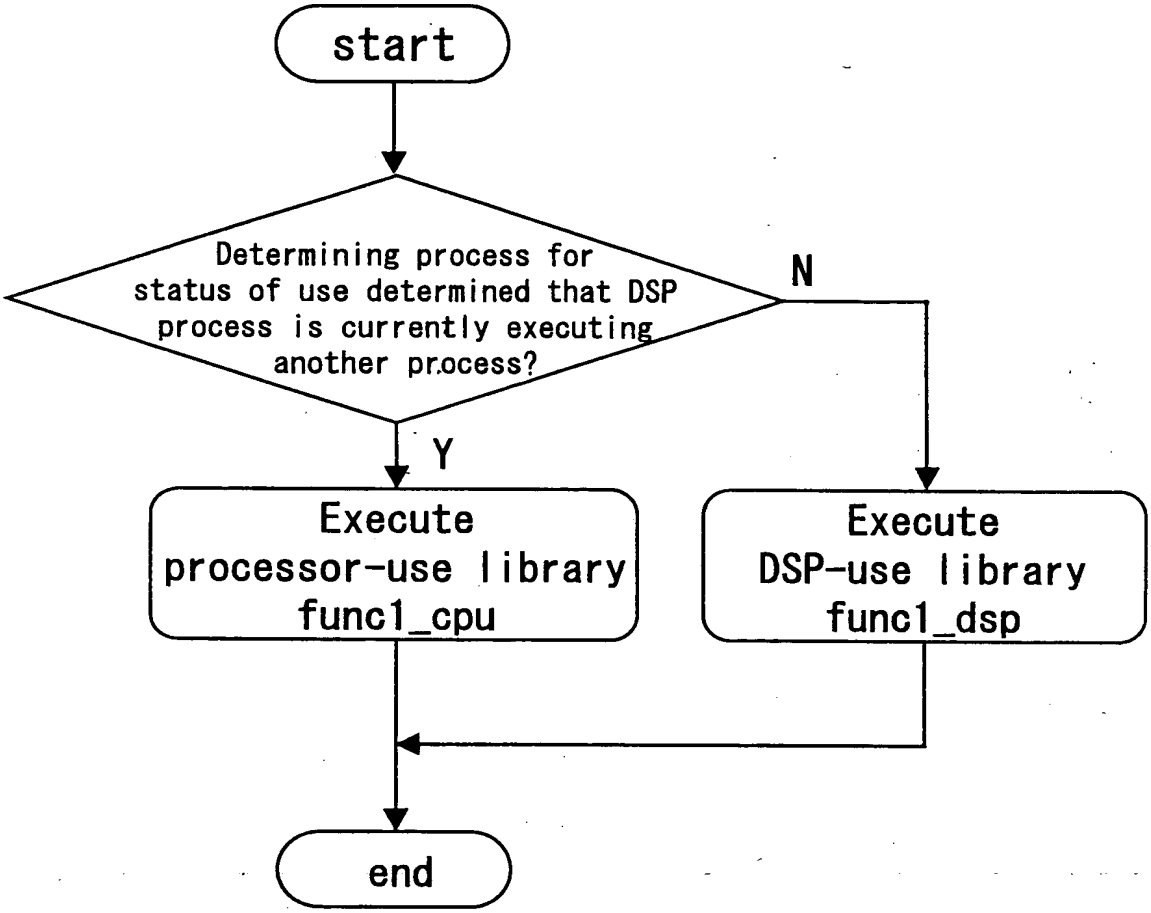

start
Determining process for status of use determined that DSP process is currently executing another process?
N
Y
Execute processor-use library func1_cpu
Execute DSP-use library func1_dsp
end

FIG. 20A

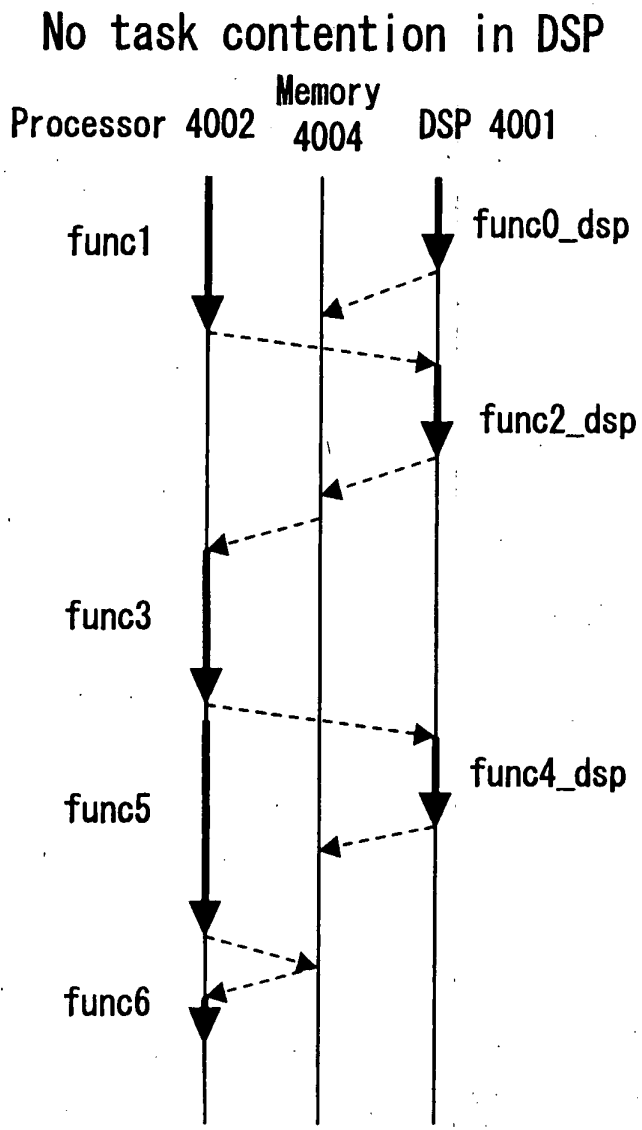


FIG. 20B

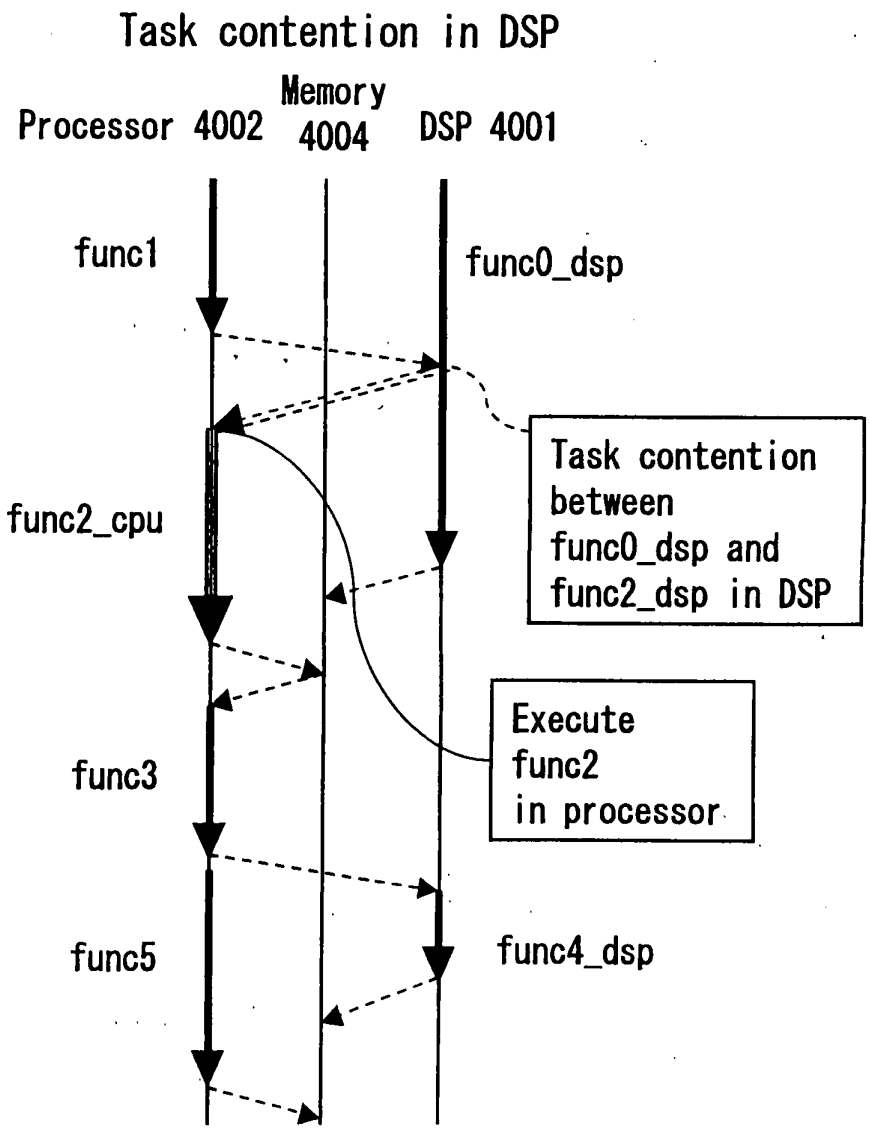


FIG. 21

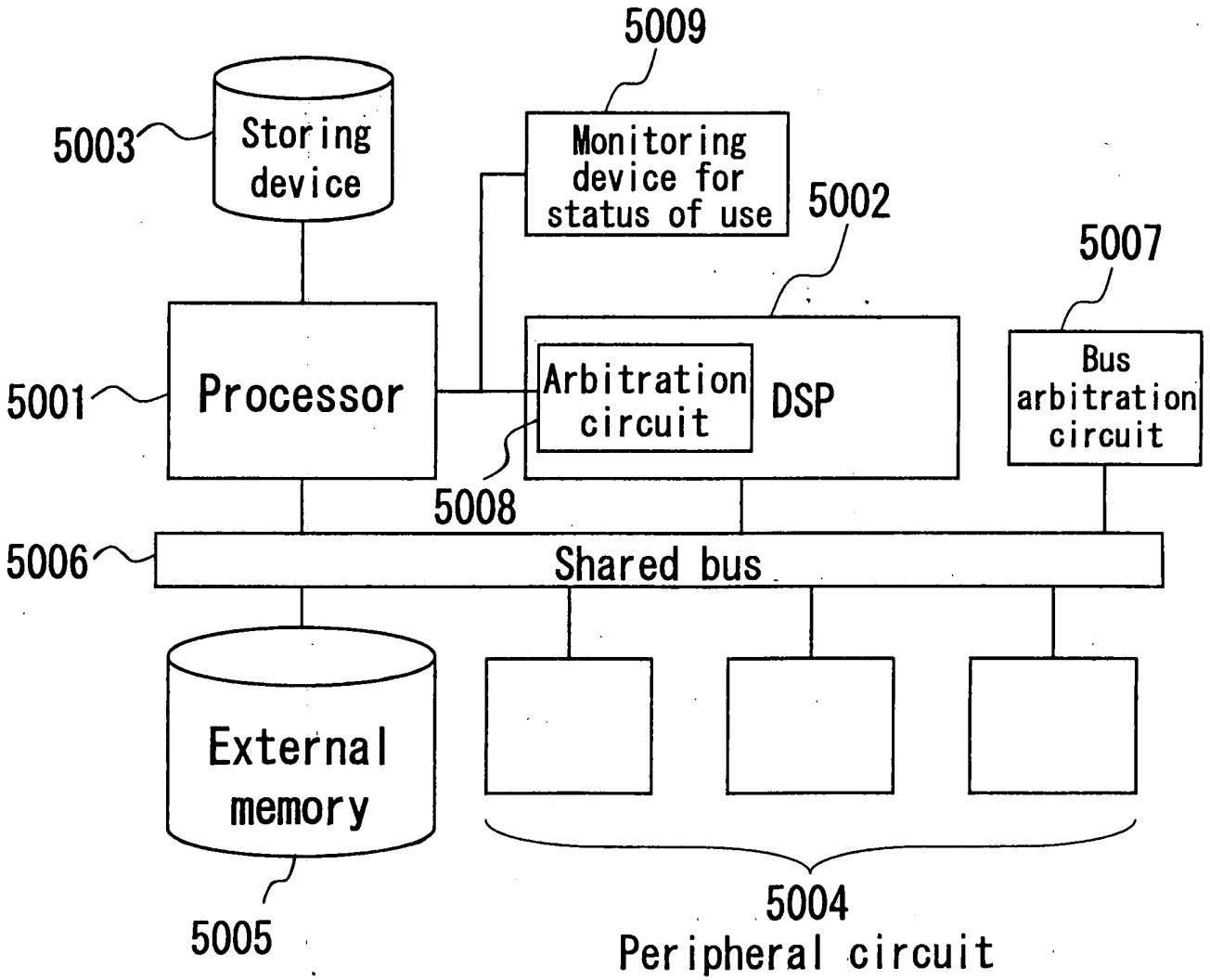

5009
5003
Storing device
Monitoring device for status of use
5002
5007
5001
Processor
Arbitration circuit
DSP
Bus arbitration circuit
5008
5006
Shared bus
External memory
5005
5004
Peripheral circuit

FIG. 22

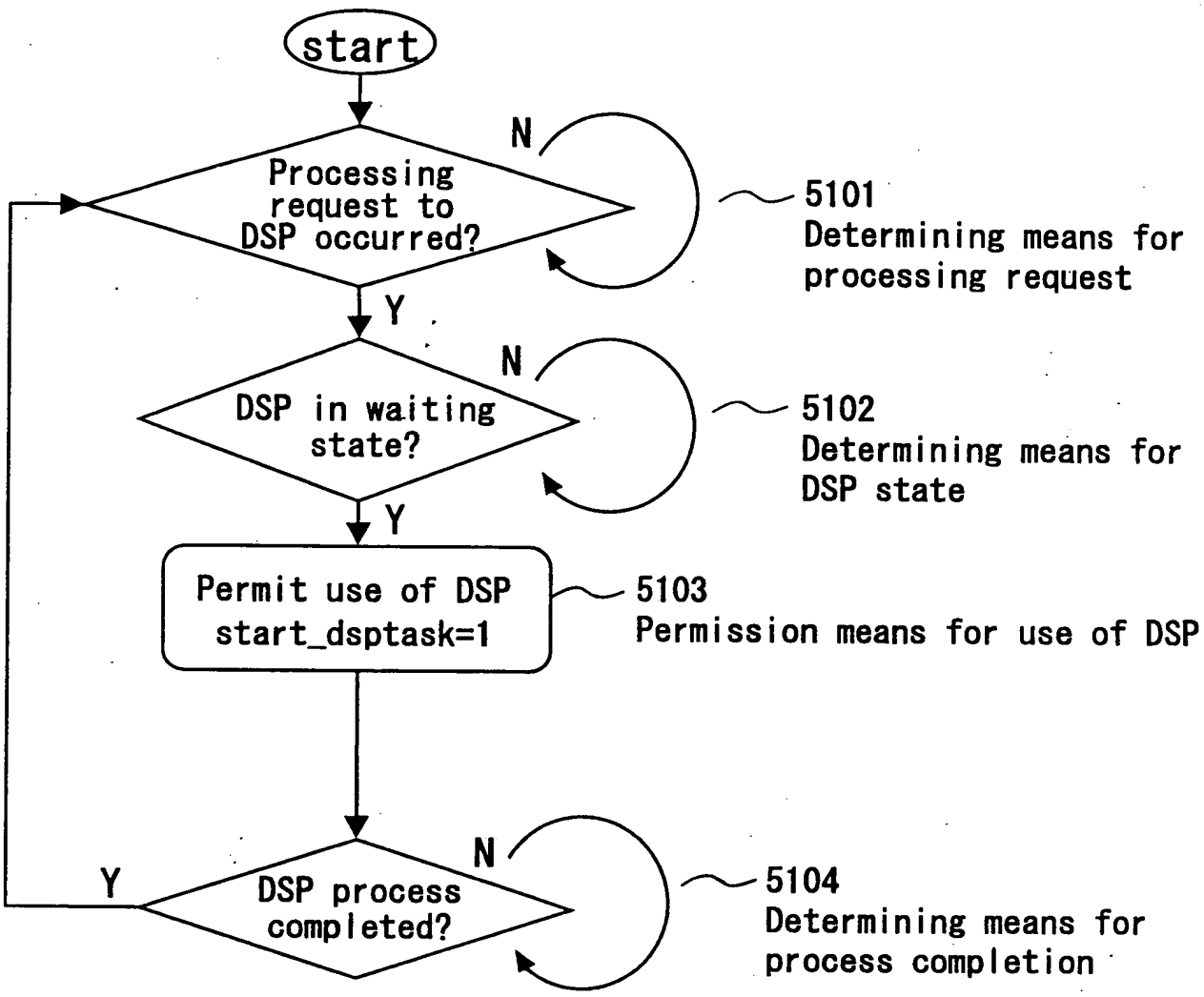

start
Processing request to DSP occurred?
N
Y
5101
Determining means for processing request
DSP in waiting state?
N
Y
5102
Determining means for DSP state
Permit use of DSP start_dsptask=1
5103
Permission means for use of DSP
DSP process completed?
N
Y
5104
Determining means for process completion

FIG. 23

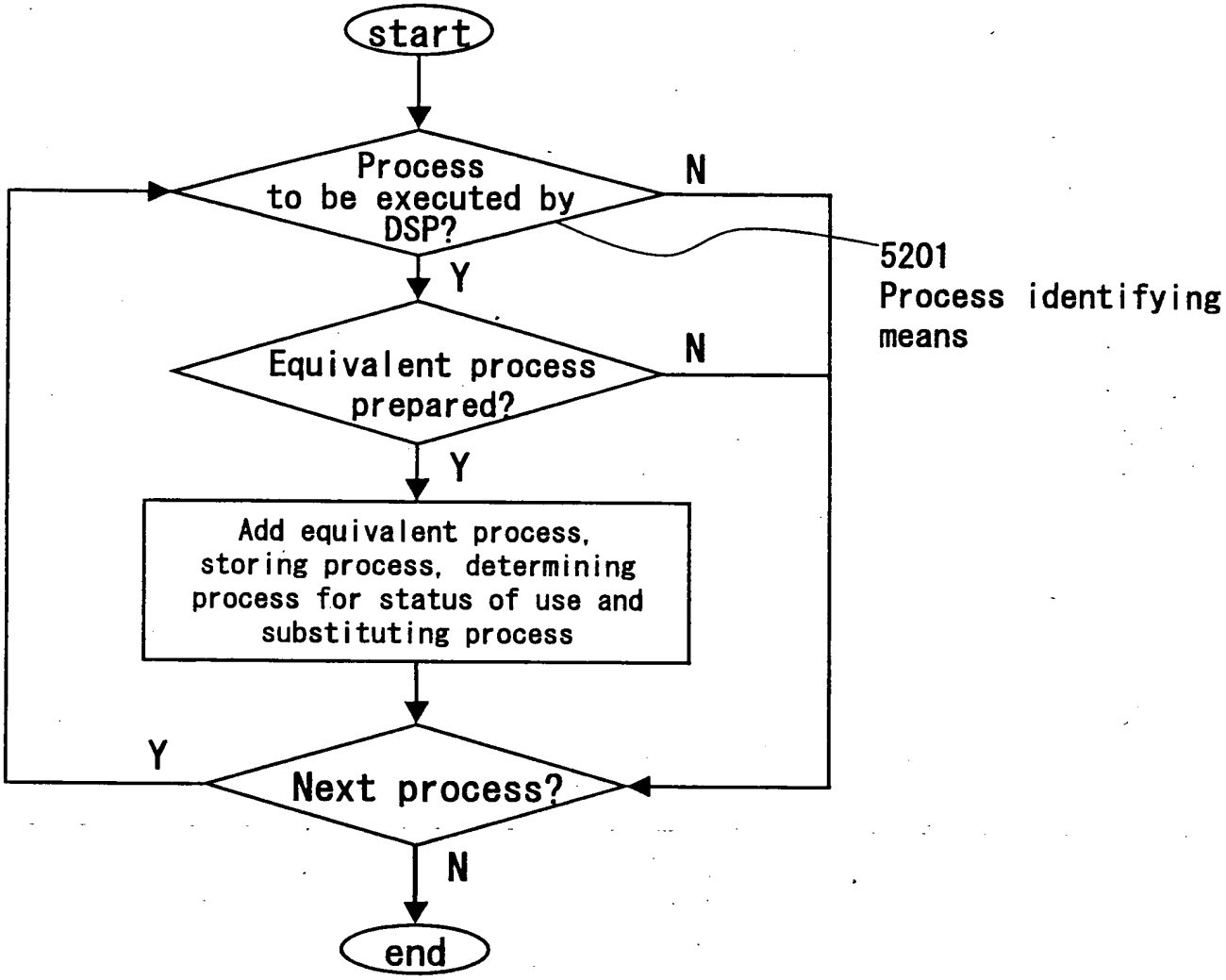

start
Process to be executed by DSP?
N
Y
5201
Process identifying means
Equivalent process prepared?
N
Y
Add equivalent process, storing process, determining process for status of use and substituting process
Y
Next process?
N
end

FIG. 24A

```
1: proc_task1(){
2:     dsp_task();
3: };
```

Prior to compiler execution

FIG. 24B

```
 1: proc_task1(){
 2:     if( wait_time < DEFINED_TIME ) {
 3:         start_time = timer_count;
 4:         Wait until DSP starts process
 5:         end_time   = timer_count;
 6:         wait_time = end_time - start_time;
 7:         dsp_task();
 8:     }
 9:     else{
10:         proc_dsptask();
11:     };
12: };
```

After compiler execution

FIG. 25

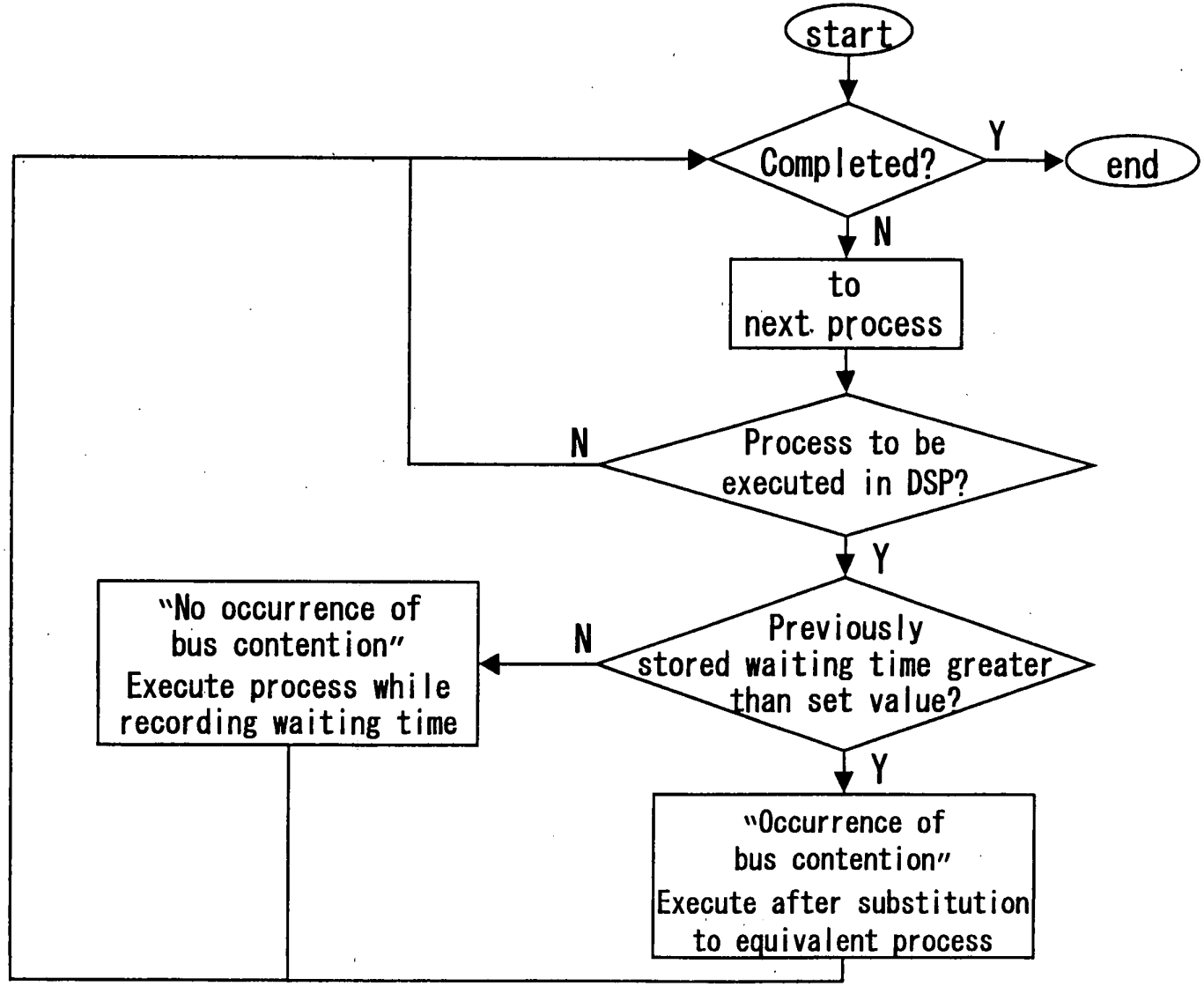

start
Completed?
Y
end
N
to
next process
N
Process to be executed in DSP?
Y
"No occurrence of bus contention"
Execute process while recording waiting time
N
Previously stored waiting time greater than set value?
Y
"Occurrence of bus contention"
Execute after substitution to equivalent process

FIG. 26

Processing time / Set value	Probability of substitution to equivalent process [%]	Probability of non-substitution to equivalent process [%]
- 2.0	60	40
2.0 - 1.8	70	30
1.8 - 1.6	80	20
1.6 - 1.4	90	10
1.4 - 1.2	95	5
1.2 - 1.0	97	3
less than 1	0	100

FIG. 27

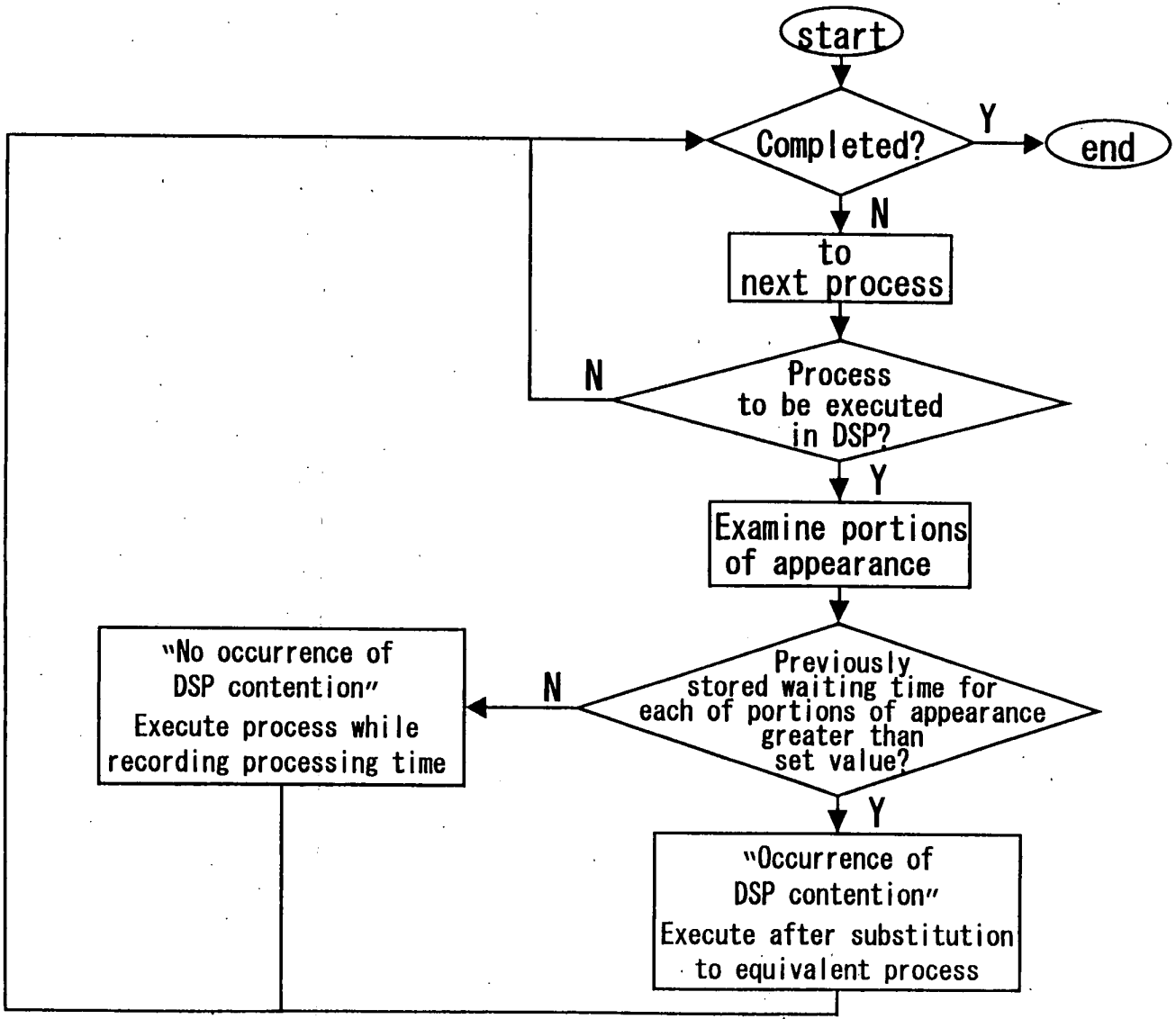

start
Completed?
Y
end
N
to
next process
N
Process
to be executed
in DSP?
Y
Examine portions
of appearance
"No occurrence of
DSP contention"
Execute process while
recording processing time
N
Previously
stored waiting time for
each of portions of appearance
greater than
set value?
Y
"Occurrence of
DSP contention"
Execute after substitution
to equivalent process

FIG. 28

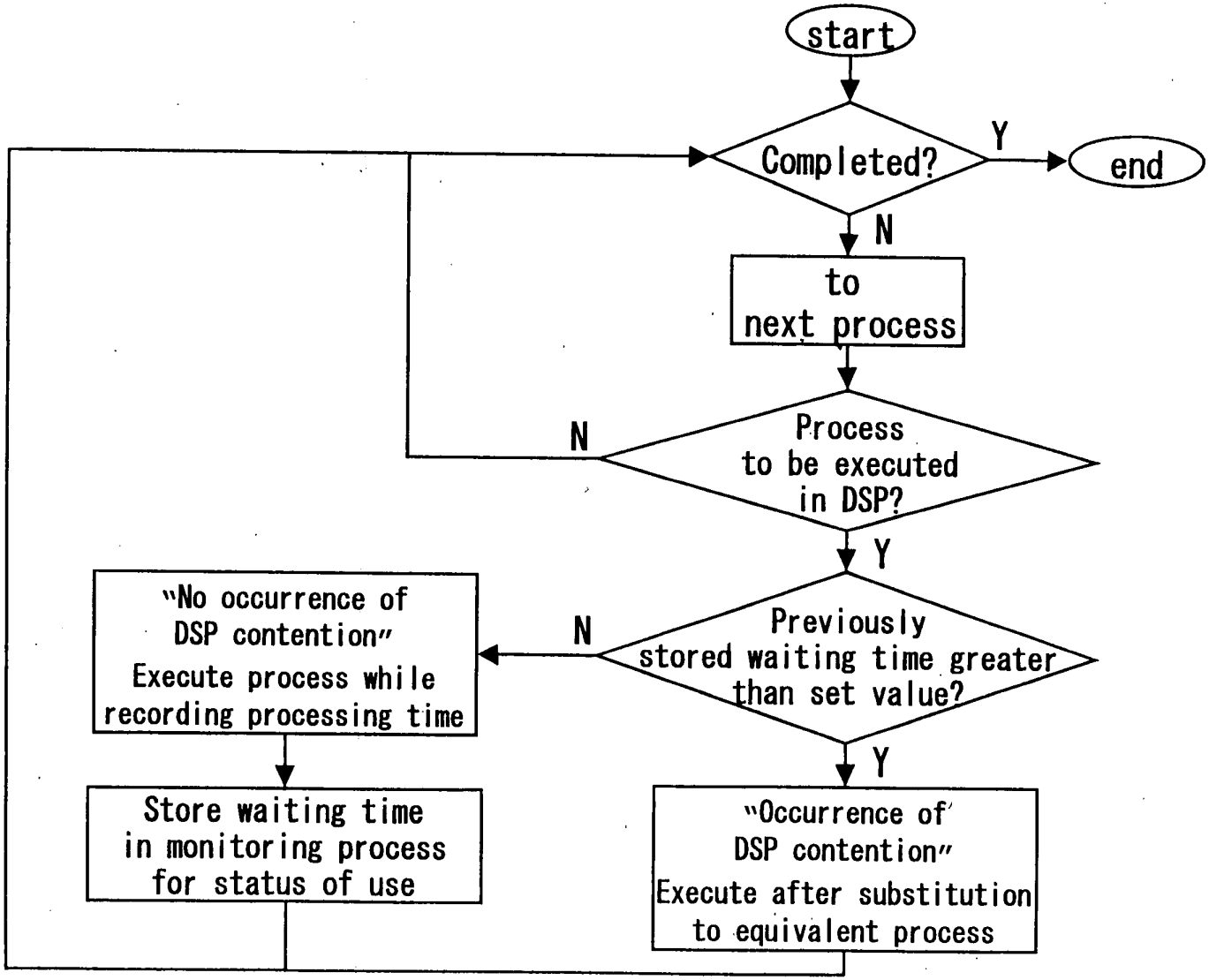

start
Completed?
Y
end
N
to
next process
Process
to be executed
in DSP?
N
Y
"No occurrence of
DSP contention"
Execute process while
recording processing time
N
Previously
stored waiting time greater
than set value?
Y
Store waiting time
in monitoring process
for status of use
"Occurrence of
DSP contention"
Execute after substitution
to equivalent process

FIG. 29

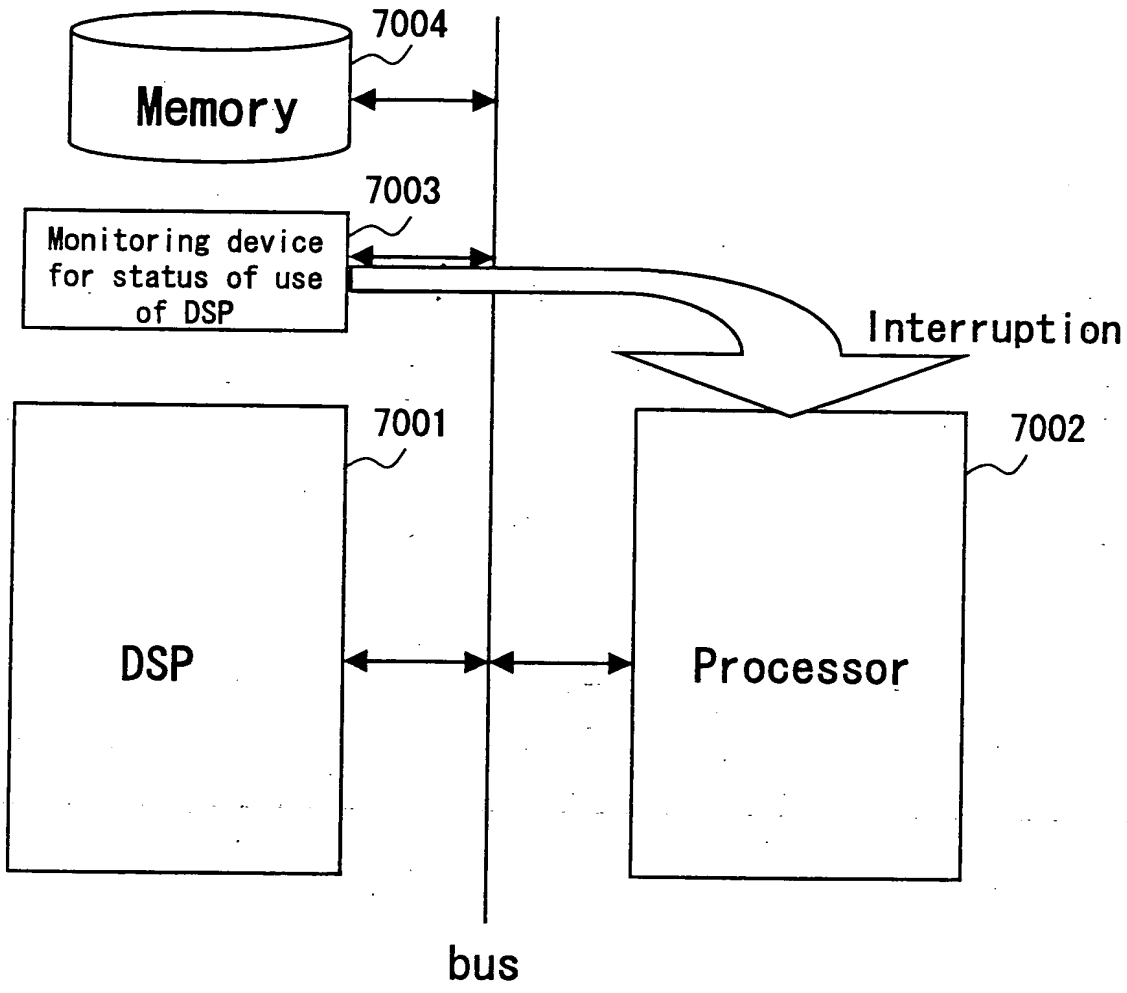

7004
Memory
7003
Monitoring device for status of use of DSP
Interruption
7001
7002
DSP
Processor
bus

FIG. 30

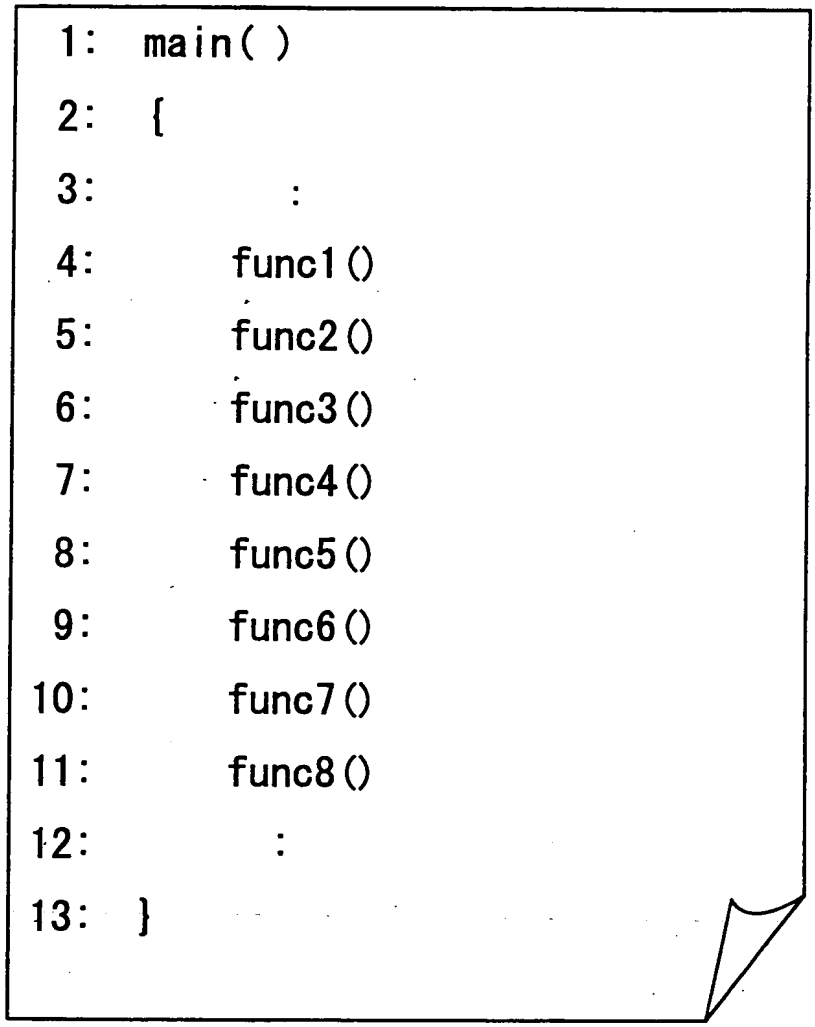

1: main( )
2: {
3: :
4: func1()
5: func2()
6: func3()
7: func4()
8: func5()
9: func6()
10: func7()
11: func8()
12: :
13: }

FIG. 31

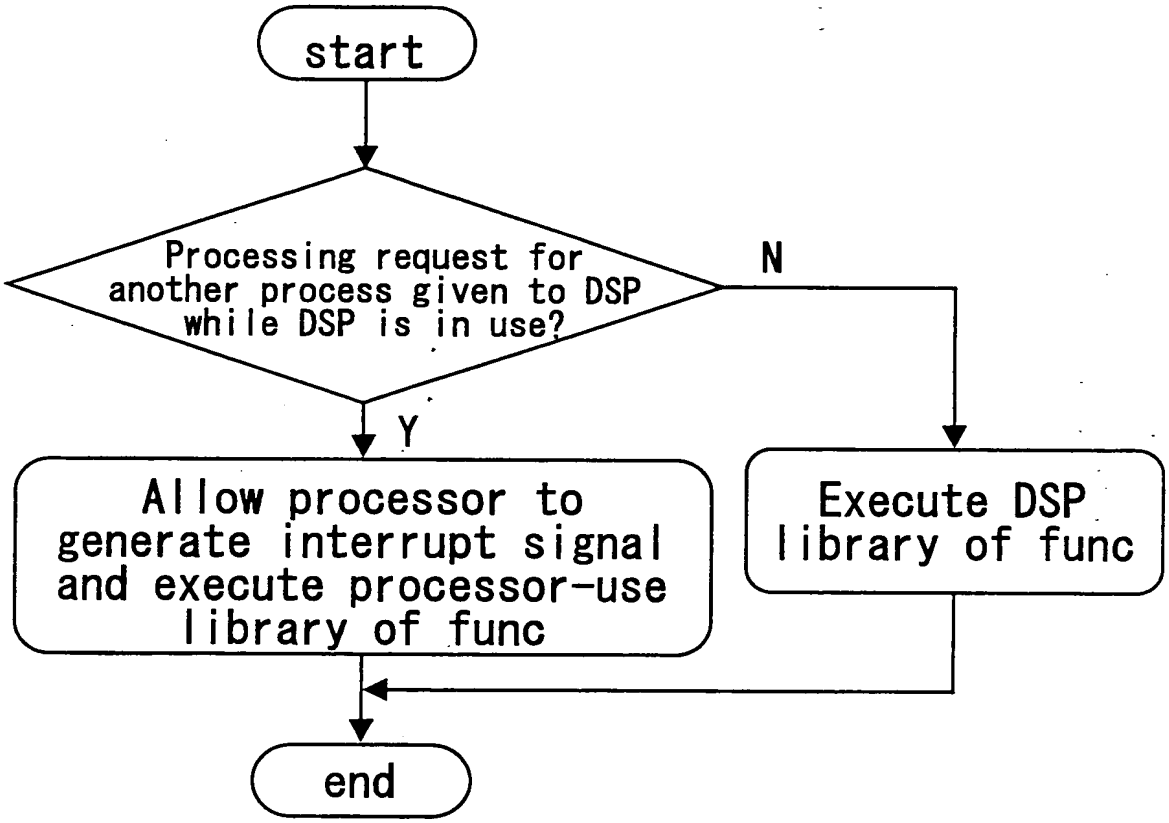

start
Processing request for another process given to DSP while DSP is in use?
N
Y
Allow processor to generate interrupt signal and execute processor-use library of func
Execute DSP library of func
end

FIG. 32A

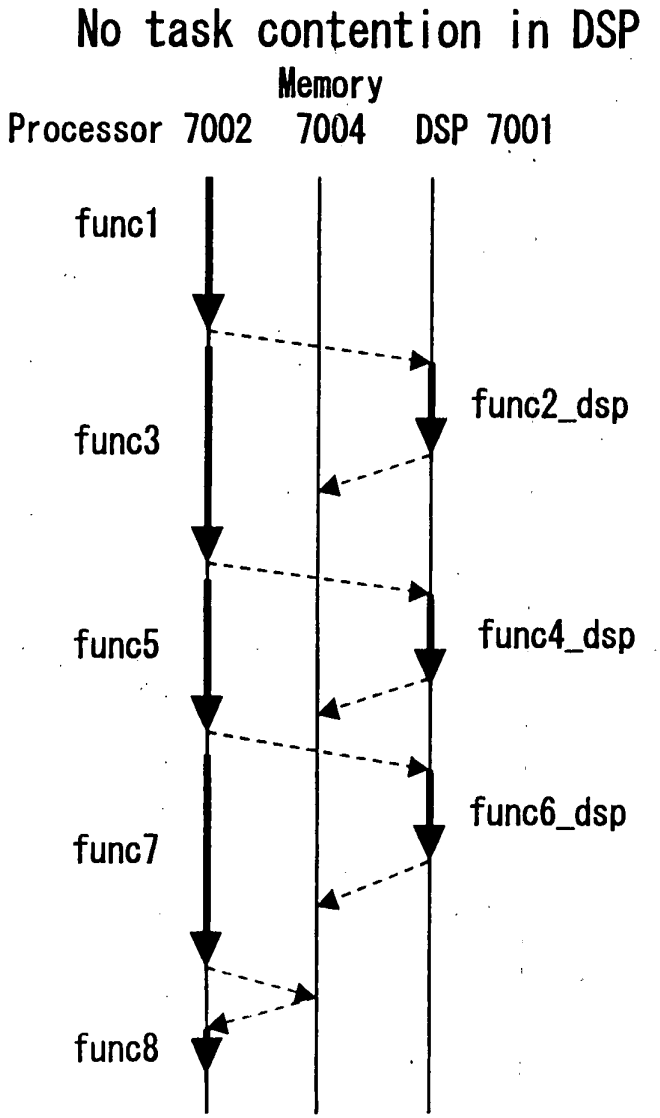


FIG. 32B

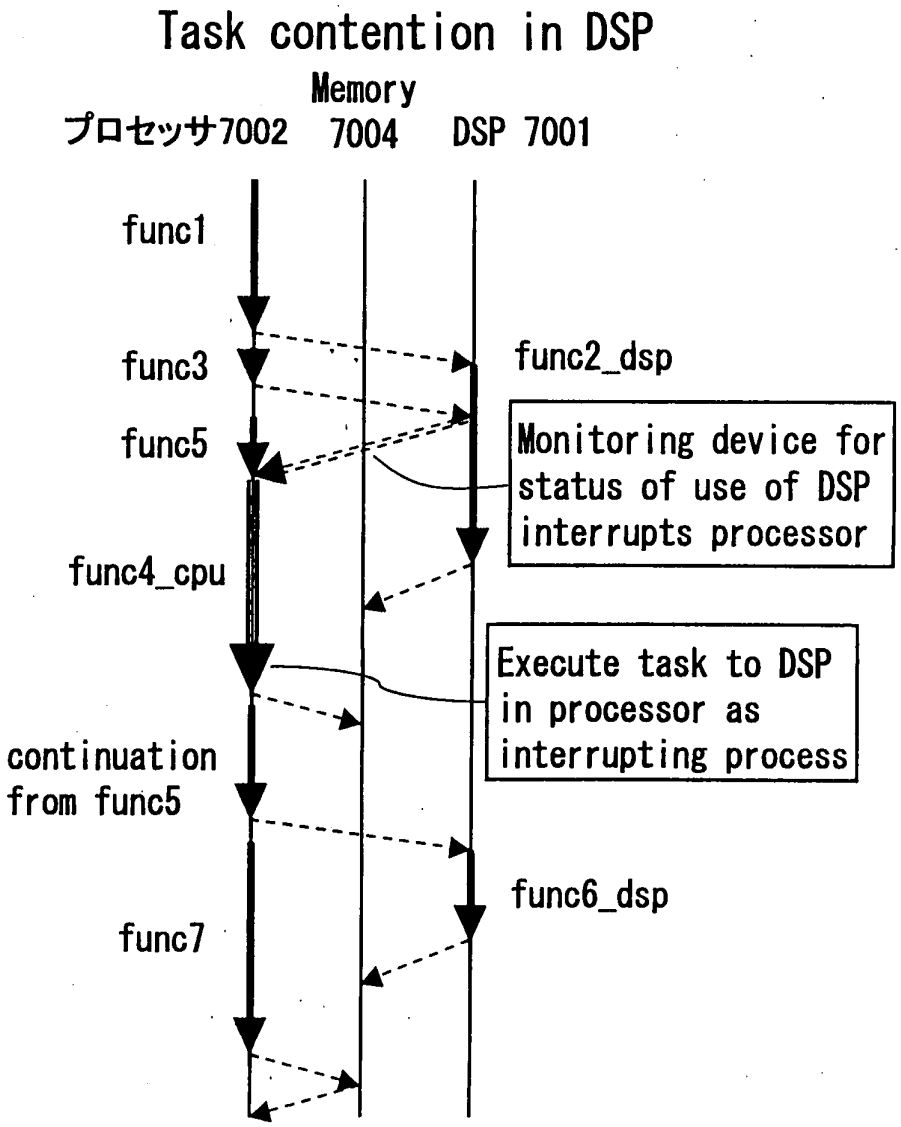


FIG. 33

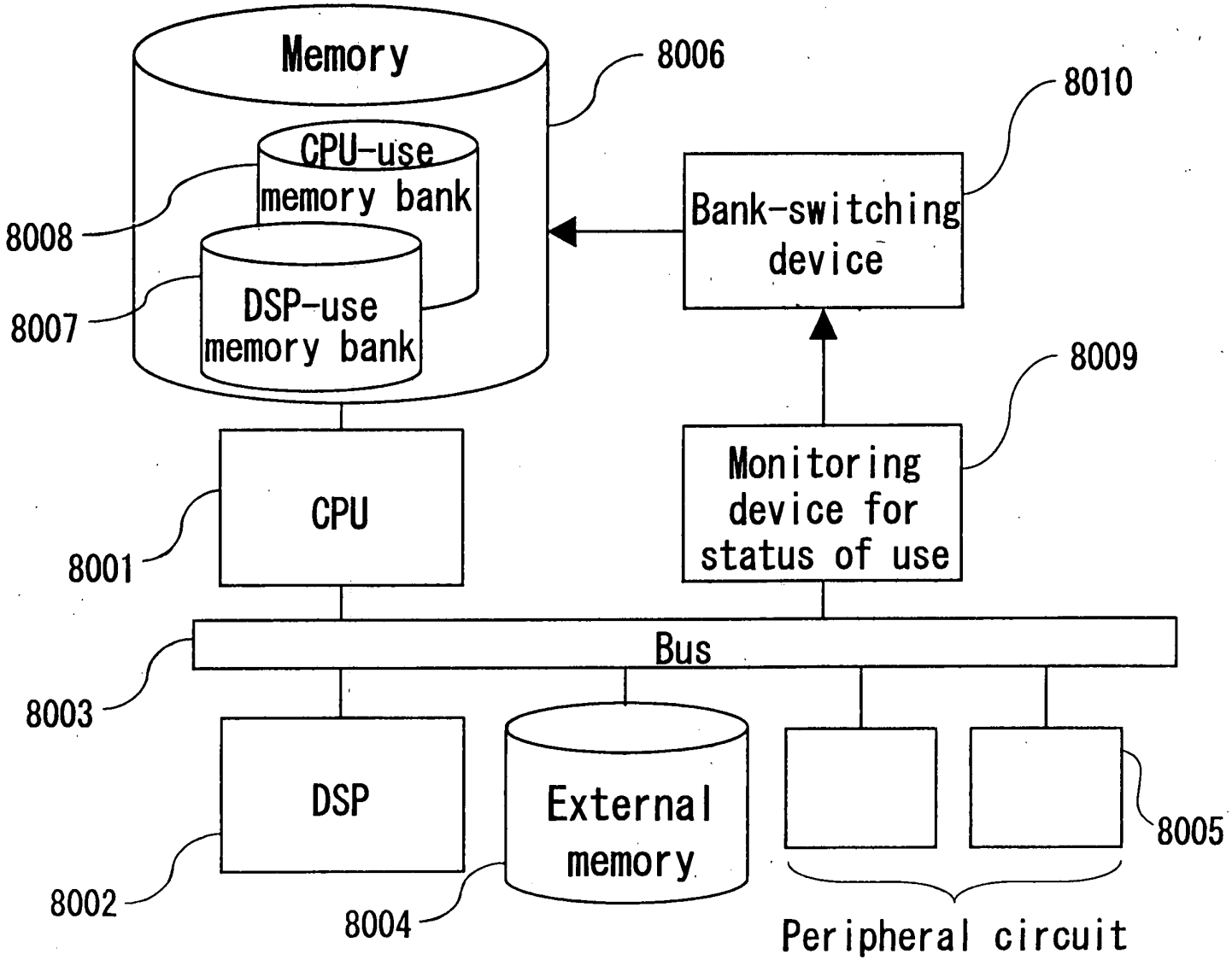

Memory
8006
8010
CPU-use memory bank
8008
Bank-switching device
DSP-use memory bank
8007
8009
Monitoring device for status of use
CPU
8001
Bus
8003
DSP
External memory
8005
8002
8004
Peripheral circuit

FIG. 34A

FIG. 34B

FIG. 34C

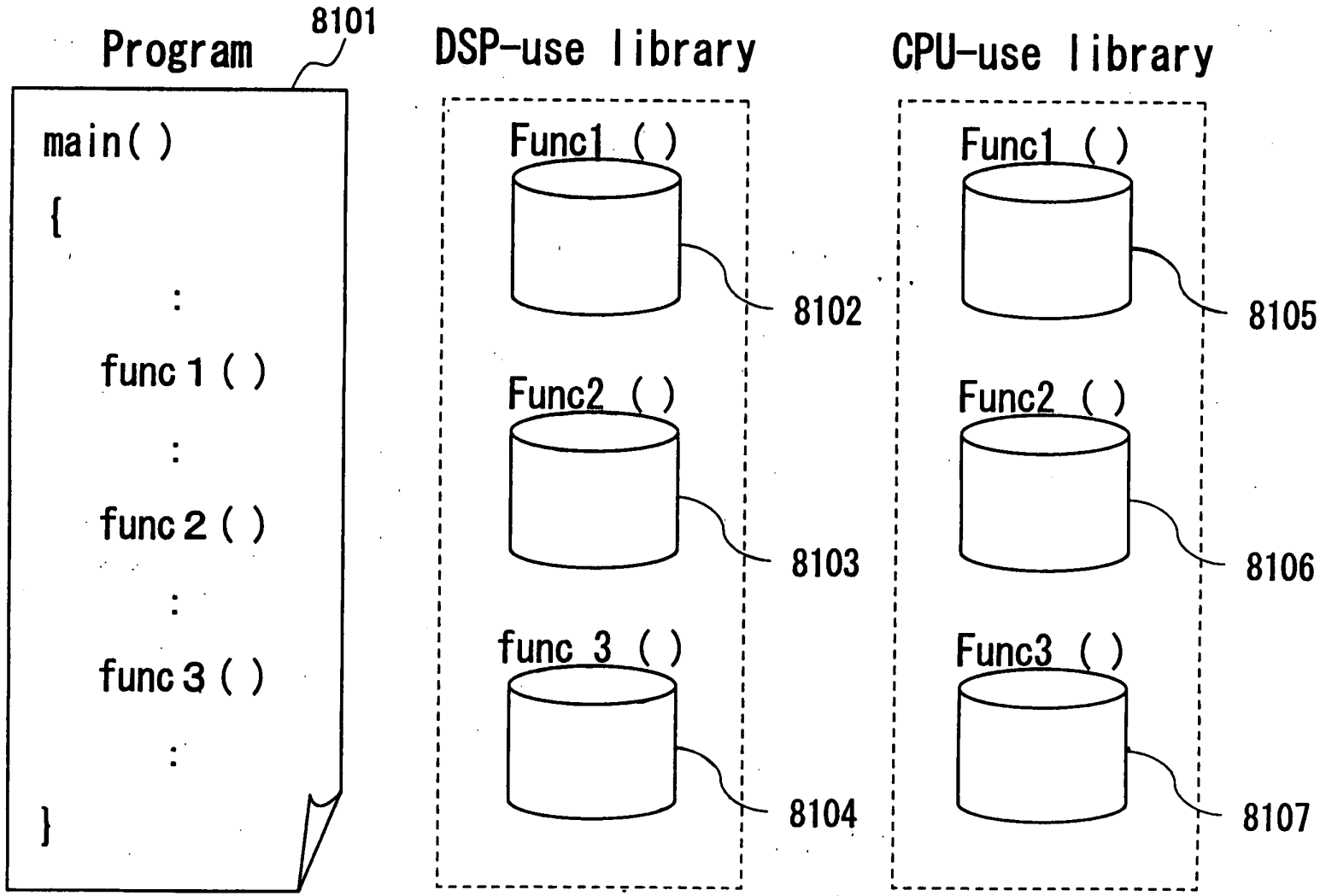


FIG. 35

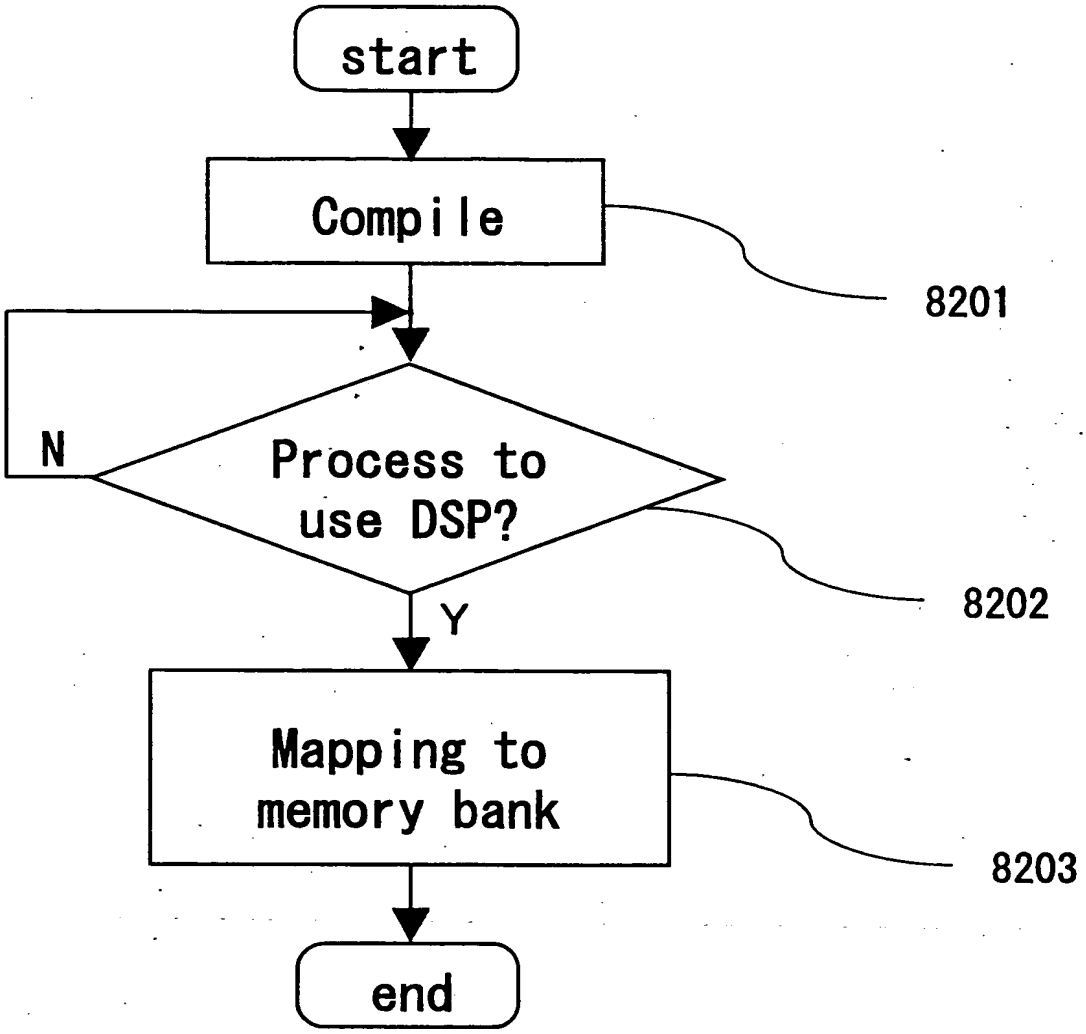
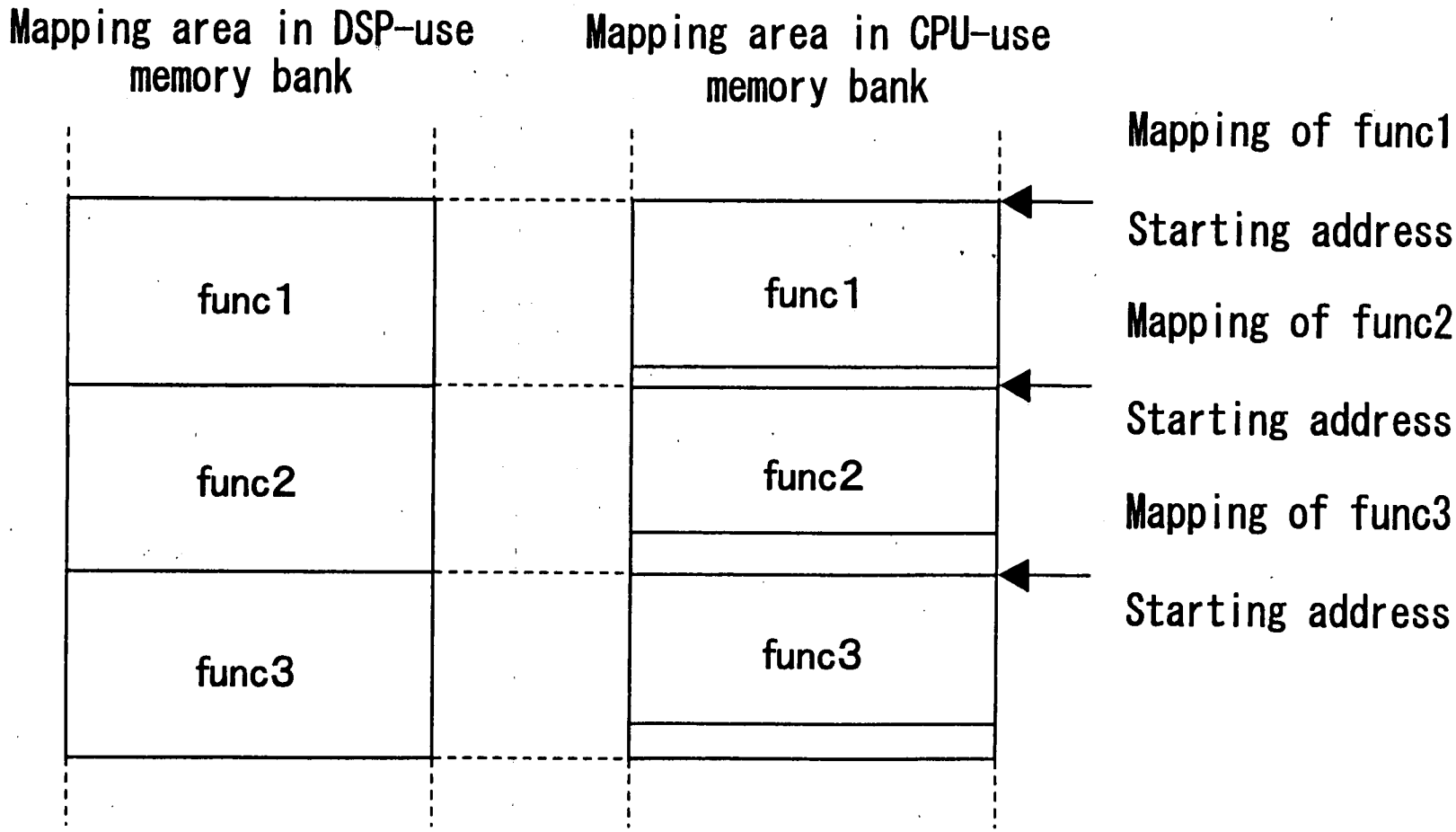

start
Compile
8201
N
Process to use DSP?
8202
Y
Mapping to memory bank
8203
end

FIG. 36


Mapping area in DSP-use memory bank
Mapping area in CPU-use memory bank
func1
func2
func3
func1
func2
func3
Mapping of func1
Starting address
Mapping of func2
Starting address
Mapping of func3
Starting address